[Advanced Search](#) [Preferences](#) [Language Tools](#) [Search Tips](#)

skew compensation "silicon graphics"

Google Search

[Web](#) · [Images](#) · [Groups](#) · [Directory](#) · [News](#)Searched the web for **skew compensation "silicon graphics"**. Results **1 - 10** of about **108**. Search took **0.30** seconds.IEEE System Packaging Workshop South Padre Island, Texas May 19 packages the system boards. Paul Everhardt of **Silicon Graphics** discusses**Skew Compensation** for the HIPPI-6400 Link. The HIPPI line is ...www.ewh.ieee.org/soc/cpmt/newsletter/199809/padre.html - 20k - [Cached](#) - [Similar pages](#)SGI - Networking GSN White Paper... Automatic **skew compensation** is an important part of the GSN link design ... it keeps thelink adjusted for **skew** and also ... Copyright © 1993-2004 **Silicon Graphics**, Inc ...www.sgi.com/peripherals/networking/gsn_whitepaper2.html - 36k - [Cached](#) - [Similar](#)[pages](#)*BIC get date/auth*[\[PDF\] 30025 SGI white paper](#)File Format: PDF/Adobe Acrobat - [View as HTML](#)... technologies supporting the GSN ANSI and IETF standards, **Silicon Graphics** is the only ... Automatic **skew compensation** is an important part of the GSN link design. ...www.sgi.com/peripherals/networking/gsn_whitepaper.pdf - [Similar pages](#)[\[PS\] Minutes of X3T11 HIPPI SWG, and HNF - Technical Committee \(TC\) ...](#)File Format: Adobe PostScript - [View as Text](#)... Greg Chesson of **Silicon Graphics** described the dynamic **skew compensation** circuitry that they were using, and said that it could compensate for upto 10 bits of ...wotug.ukc.ac.uk/parallel/standards/hippi/minutes/feb96_hippi_min.ps - [Similar pages](#)*Have
ELL open this
document*TUTORIALS... Course Material - Multistage opamp **compensation** root loci ... gating, **skew** management, **skew** budgeting, latch ... a circuit designer at **Silicon Graphics**, Mountain View ...www.isscc.org/isscc/1998/ap/TUTORIALS.htm - 8k - [Cached](#) - [Similar pages](#)ISSCC Tutorials List... and nested Gm-C **compensation** are described ... Instructor: John Maneatis, at **Silicon Graphics**, Mountain View ... clock distribution, clock gating, **skew** management, **skew** ...www.isscc.org/isscc/ISSCCTutorial.htm - 24k - [Cached](#) - [Similar pages](#)[\[More results from www.isscc.org \]](#)TechOnLine - Digital vs. Analog Design for a Multi-Gigabit SerDes... a process-voltage-temperature (PVT) **compensation** circuit is ... process sensitivity, testability, and **skew** management, a ... He worked at **Silicon Graphics** as a system ...www.techonline.com/community/tech_group/analog/31364 - 42k - [Cached](#) - [Similar pages](#)HIPPI-6400 OVERVIEW... The 10 ns **skew compensation** is required at the receiver ... Current worst case fiber-to-fiber **skew** is set ... **Silicon Graphics** has contracted to both groups to build ...www.lanl.gov/lanp/hippi64/papers/spy.html - 34k - [Cached](#) - [Similar pages](#)[\[PDF\] A 2.4 Gb/s/pin simultaneous bidirectional parallel link with per ...](#)File Format: PDF/Adobe Acrobat - [View as HTML](#)

... 11, NOVEMBER 2000 1619 A 2.4 Gb/s/pin Simultaneous Bidirectional Parallel Link with

Per-Pin **Skew Compensation** Evelina Yeung, Student Member, IEEE, and Mark A. ...
mos.stanford.edu/papers/ey_jssc_00.pdf - [Similar pages](#)

[EEDesign.com](#) - Exploring new design flows -- physical layout
... windows and dynamic coupling **compensation** coefficients to ... place and route, **skew**
clock generation ... Adobe Systems, Hewlett Packard, **Silicon Graphics**, Rohm, and ...
www.eedesign.com/story/OEG20020628S0107 - 75k - [Cached](#) - [Similar pages](#)

Google

Result Page: 1 2 3 4 5 6 7 8 9 10 [Next](#)

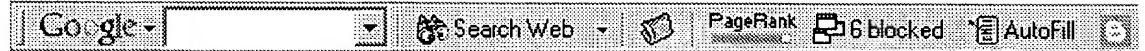
skew compensation "silicon graphics"

Google Search

[Search within results](#)

Dissatisfied with your search results? [Help us improve.](#)


Get the [Google Toolbar](#):



[Google Home](#) - [Advertise with Us](#) - [Business Solutions](#) - [Services & Tools](#) - [Jobs, Press, & Help](#)

©2004 Google

Gigabyte System Network GSN Technical Overview

 **SifconGraphics**

Document Summary

File: C:\WINNT\Profiles\sgreen\Desktop\gsn_whitepaper.pdf

Title: B0025 SGI white paper

Subject:

Author: Steve EPP

Keywords:

Binding: Left Edge

Creator: QuarkXPress®: LaserWriter 8 8:3:4

Producer: Acrobat Distiller 3.01 for Power Macintosh

Created: 11/5/98 9:53:56 AM

Modified: 11/5/98 9:55:21 AM

File Size: 53.5 KB (54,764 Bytes)

Security: None

PDF Version: 1.2 (Acrobat 3.x)

Page Size: 8.5 in x 11 in

Number of Pages: 8

Fast Web View: No

Tagged PDF: No

OK

Gigabyte System Network GSN Technical Overview

Gigabyte System Network Technical Overview

Gigabyte System Network, or GSNTM, is the highest bandwidth and lowest latency interconnect standard, providing full duplex 6400MB per second (800MB per second) of error-free, flow-controlled data transmission. The technology is ideal wherever organizations require timely and efficient movement of large amounts of information, including scientific and technical computing, HDTV, data mining, transaction processing, video and film archiving, and storage management. The proposed ANSI standard provides for interoperability with Ethernet, Fibre Channel, ATM, HIPPI-800, and other standards.

For technical computing applications such as clustering and system area networks and for enterprise applications of big data client-server functions (e.g. HDTV, post-production scanners, MR medical imaging) and storage management backbones that need huge bandwidth, low latency, and extremely efficient CPU utilization, Silicon Graphics GSN is the only networking solution that can provide it all with an outstanding throughput price/performance value. As the key developer of the enabling technologies supporting the GSN ANSI and IETF standards, Silicon Graphics is the only computer vendor to introduce GSN before the end of the century.

Gigabyte System Network (GSN), also known as HIPPI-6400, is a physical layer (PHY) currently being developed within the HIPPI family of interconnect standards. The official standard number will be ANSI NCITS 323-199x. This paper is designed to provide an overview of the design; however, many details of GSN are beyond the scope of this overview. Examples include error processing, link reset and initialization, training sequence waveforms, crc equations, complete protocol algorithms or state machines, IP encapsulation, bridging algorithms, and details of HIPPI-800 compatibility. The interested reader is invited to the HIPPI Standards Activities Web page at www.cic-5.lanl.gov/lanp/ANSI/.

Introduction

The HIPPI link technology most commonly used today is HIPPI-800, a 32-bit simplex parallel interface defined for copper cables clocked at 25 MHz. The raw bandwidth of HIPPI-800 is 800MB per second. GSN defines a 20-bit interface for copper cables operating at 500 MHz, or a 10-bit interface for fiber-optic cables operating at 1 GHz. It has a raw bandwidth of 10000MB per second in each direction. This provides a payload bandwidth of 6400MB per second in each direction after subtracting the overheads of ac-encoding and control information.

The GSN effort began with a combination of circumstances: a desire for gigabyte per second networking compatible with existing HIPPI and Ethernet networks, proof-of-concept technology for high-bandwidth links and routers (#1), and the ability of the ANSI T11 standards group to consider another PHY for HIPPI.

Interconnect Traffic Profiles

Computer interconnects carry a wide variety of traffic types because of the broad set of applications that utilize distributed computing resources. Examples include MPI-based parallel computation, file serving, Web serving, datamining, transaction processing, video and image archiving, and distribution. This is certainly a broader range of applications than the telnet-ftp-mail scenario.

The traffic generated by these applications may vary from small messages of a few bytes to large multi gigabyte bulk transfers. In addition to the size metric one can also distinguish different message frequency distributions ranging from asynchronous to bursty to continuous. There may also be hard real-time components where a priority discipline or a completion time discipline is needed.

The proliferation of applications and their different traffic profiles has encouraged the development of optimizations. Modern architectures such as ATM and Fibre Channel represent a departure from older designs such as the IEEE 802.3 Ethernet family in a significant way: these designs attempt to explicitly accommodate a wide variety of traffic types.

ATM approaches the problem with several techniques: (1) a common cell structure with numerous adaptation layers (AAL) that are specialized framing procedures, plus (2) explicit quality-of-service parameters and means for administering them (e.g. rate control), and (3) explicit procedures for constant-bit-rate (CBR) traffic. The Fibre Channel standard approaches the issue of different traffic types by defining different classes of service and several physical media. Both ATM and Fibre Channel exceed the scope of Ethernet and HIPPI-800 by a wide margin. They also exceed the complexity of the other designs by a wide margin.

Objectives

The main performance objectives for GSN are high bandwidth and low latency: 800MB per second transfers and 1 microsecond latency for short transfers over short distances. Other objectives include:

1. Reliable, flow-controlled links
2. OS Bypass support
3. Multiplexing of messages on a link
4. Compatibility with HIPPI-800

Flow control in GSN depends on a credit-based protocol. Having reliable links means having some sort of error detection and retransmission or possibly forward error correction. GSN uses 32-bits of crc for error detection over a 32-byte micropacket plus a sliding window protocol that supports retransmission of damaged macropackets.

Flow controlled links leading into a switch can be used by the switch to eliminate congestion within the switch. A fabric that is both reliable and flow controlled can interface directly to an application without going through the usual operating system and protocol stack software layers.

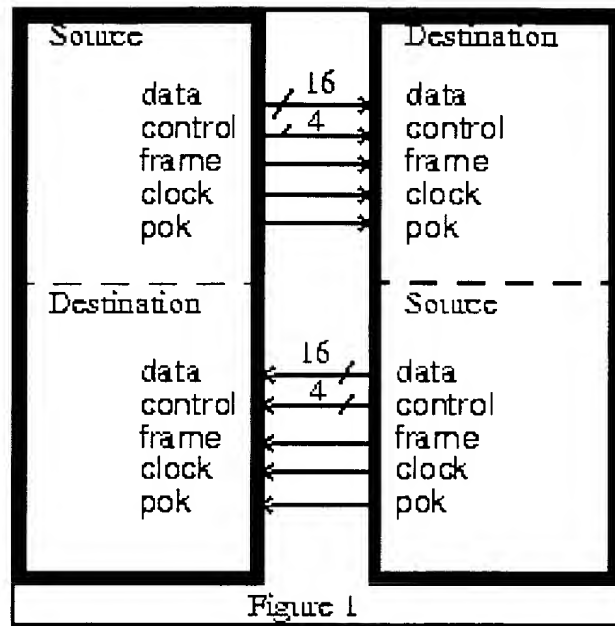
Operating system bypass support means moving data between a link and an application without OS system calls or intervention. A new upper layer protocol called the scheduled transfer protocol defines a standard method of OS bypass.

Other HIPPI PHY layers do not multiplex. This means that while one system is connected to a destination system, others must wait for that destination. Since HIPPI messages may be arbitrarily long, it is not possible to bound the waiting time. GSN provides four virtual channels (VCs) so that up to four messages can progress across a link at the same time. Maximum message sizes are also specified so as to bound the busy time on each VC. Data transfers larger than the desired message size are broken into smaller blocks for transmission. The details of decomposing a large transfer and collecting the blocks within an application buffer are controlled by the schedule header.

Compatibility with HIPPI-800 is maintained by defining a representation within GSN for all the packet types defined for HIPPI-800 and by defining procedures for implementing a transparent bridge between the two PHYs. The use of 48-bit IEEE 802.3-style addresses allows bridges to other datalink protocols to be implemented.

GSN Link

There are two kinds of media defined for GSN: copper coaxial and optical fiber. The copper link is based on a 50-pair coaxial cable and 100-pin connector. The signals are defined in Figure 1. The diagram shows 16 data and four control lines in each direction along with a source-synchronous clock, frame indicator and power_ok (pok) indicator. These add up to a total of 46 signals, or 92 pairs, leaving eight wires for shield connection, grounding, and supplying a small amount of power to an external device. The parallel link is thought of as a 16-bit payload clocked at 400MHz for a bandwidth of 6400MB per second. However, the data is encoded with a 4b/5b code and sent at 500MHz.



The optical media to be selected for GSN depend on the continued evolution of parallel optical fiber components. At this time the proposed optical link differs from Figure 1 in that there are eight data lines in each direction instead of 16; there are two control signals instead of four. The frame and clock signals are retained, and the pok signals are not needed. The total number of signals in each direction is 12. The clock frequency is 1GHz using the same 4b/5b code defined for coaxial.

GSN link protocol is designed to accommodate links up to 1 km in length. The propagation time for optical fiber is roughly 5 ns/m, so the propagation time for a 1 km link would be 5 microseconds. Five microseconds of data at 800MB per second is 4000 bytes. Thus an 8KB (8192) transmission buffer is sufficient to cover the round-trip delay of a 1 km GSN link with a little spare time.

Automatic skew compensation is an important part of the GSN link design. Without it a simple parallel link of this kind would not be able to cover much distance: skew buildup could easily exceed the clock period of 2 ns. The skew buildup in optical ribbon cable varies from a low of about 1.5 ps/m to as much as 10 ps/m. Thus skew for a 1 km link could vary between 1.5 and 10 ns, depending on the quality of the cable. GSN compensates for skew by defining a special bit pattern called the training sequence which is used at the receiving end of a link to sense the amount of skew on each signal line. Receiver circuitry can use this information to compensate for skew. The training sequence is used at link startup time and is also invoked at periodic intervals (10 microseconds). The frequent retraining interval does two things: it keeps the link adjusted for skew and also injects a sufficient periodic gap into the flow of data to compensate for relative clock drift between the sending and the receiving endpoints.

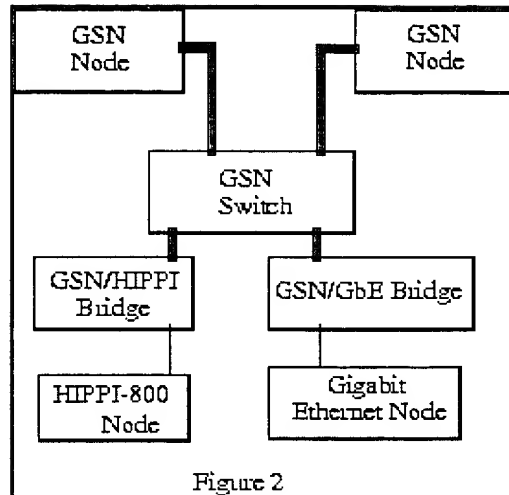
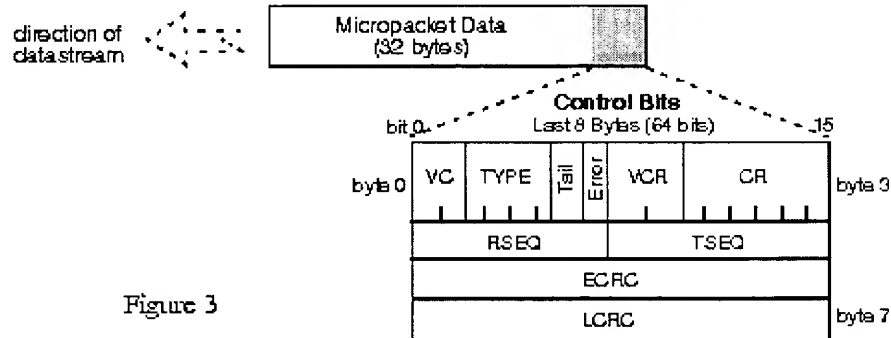


Figure 2 illustrates the link-oriented view of GSN: the thick lines each represent a GSN link. The links are shown connecting GSN nodes and transparent bridges to a GSN switch. The bridges provide transparent bridging to HIPPI-800 and Gigabit Ethernet. Bridges can be designed to accommodate other media that also use 802-style frame headers.

Micropackets and Messages

GSN defines a micropacket as the smallest unit of data transfer. Each micropacket consists of a 32-byte data payload together with a 32-bit control word. As implied by the cable design, the control word is transmitted in parallel with the data bytes. The control word is organized as shown below.

The bitfield definitions are as follows:



- CR--6-bit flow control credit field
- VCR--specifies VC to receive credit
- E--error bit
- T--tail bit (last micropacket of a message)
- TYPE--micropacket type
- VC--specifies virtual channel
- TSEQ--transmit sequence number
- RSEQ--received sequence number
- ECRC--end-to-end CRC
- LCRC--link CRC

GSN defines an 8-bit up/down credit counter for each of the four virtual channels. The counter is cleared on link initialization or reset. A received micropacket can bestow a number of credits as indicated in the CR field to the counter indicated by VCR. Credits are decremented when sending -- one micropacket equals one credit -- and incremented when credits are received.

Each micropacket contains a pair of 8-bit sequence numbers, TSEQ and RSEQ, which form the basis for a sliding window protocol. Output micropackets are numbered via TSEQ and are acknowledged via RSEQ. The receiver discards micropackets that do not contain the expected TSEQ value, which is monotonically increasing. A transmitter will resend micropackets if they are not acknowledged within an (adjustable) interval.

Each micropacket is associated with one of the four virtual channels as specified by the VC field. A GSN message is defined as a sequence of micropackets consisting of a header micropacket followed by one or more data micropackets. These and other micropacket types are indicated by the TYPE field in each control word. The last micropacket of a message is indicated by the TAIL bit.

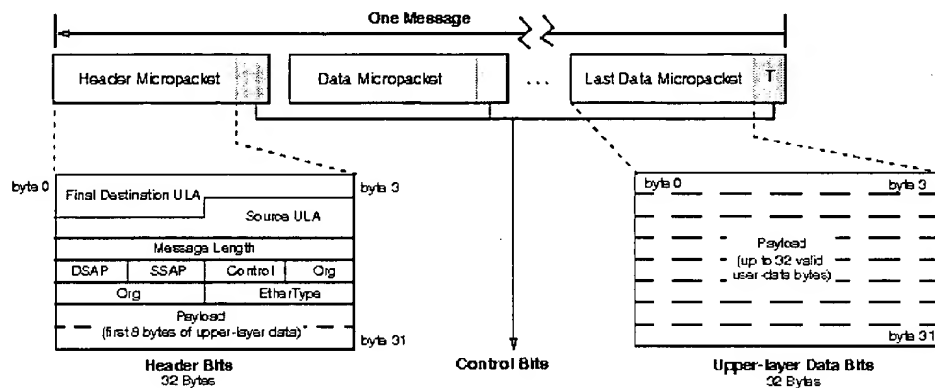


Figure 4 GSN Message Composed of Header and Data Micropackets

Figure 4 illustrates the organization of a message as an ordered sequence of micropackets. A link or any GSN switching device is required to preserve the order of micropackets within each virtual circuit, although links and switches are allowed to alternate between virtual channels to ensure fairness.

Each micropacket is protected by a pair of 16-bit crc functions. The LCRC is calculated over the bits of each micropacket excepting the LCRC itself. The ECRC is calculated over the data payload. The ECRC transmitted with each data micropacket of a message is the accumulated ECRC over all the preceding micropackets within the message.

The ECRC is called the end-to-end crc because it is generated once when a message is created and passed without change through switches and links until the message arrives at the final destination. Although the ECRC may be checked at intermediate points as well as the final destination, it is never modified or recalculated. Where the ECRC is checked, it is tested on every 32-bit micropacket. This scheme reduces the probability of an undetected error in a long message to the same probability as a 32-byte message. If either crc detects an error, the most common action is to discard the micropacket. (#2).

Exactly one message is allowed to be in progress on each virtual channel of a GSN link. However, micropackets from all four VCs may be multiplexed on the link at the same time. An obvious use of this facility is for separating bulk traffic and low-latency traffic onto separate VCs. Another side-effect is that a receiver must provide sufficient buffering for each VC to maintain a full-bandwidth link: i.e. 256 micropackets. The combination of a small number of VCs plus the 8-bit sequence numbers results in exact numbers for the amount of buffering needed to operate a link. The total amount of receiver buffering needed is 32 KB.

A single 8K send buffer is also required. It is practical to place this amount of memory on an ASIC. Thus, a full-performance GSN switch port or computer interface can be constructed without using dedicated buffer memory chips external to a MAC device.

All the micropackets of a message are carried between endpoints on the same virtual channel. There is a specific assignment of traffic to virtual channel according to message size. Small messages less than 2KB in length are assigned to virtual channel 0 (VC0). Messages less than 128KB may be sent on VC1 or VC2. VC3 may be used only for scheduled transfers as discussed in the next section. The other VCs may be used for scheduled transfers if desired. This arrangement allows small messages to progress without experiencing queuing delays behind long messages.

Universal LAN MAC Addresses

ULAs, or universal LAN MAC addresses, are 48-bit IEEE 802.3-style addresses. Every GSN device will be assigned a ULA. When a GSN device is first enabled and connected to a GSN link, the device knows its ULA and little else. Procedures are defined in GSN whereby the device can determine whether it is connected to a switch or to another device. If connected to a switch, additional procedures are defined for the switch to assign a unique MAC address to the GSN device.

GSN also provides a broadcast message capability sufficient to support the ARP protocol. ARP and proxy-ARP provide the basis for building transparent bridging devices that attach to GSN. Although a detailed discussion is beyond the scope of this paper, the destination ULA (D-ULA) and source ULA (S-ULA) fields are provided in the MAC header for the use of translators of the kind depicted in Figure 2.

Header Structure

As described in section 5.0, the mechanisms that implement flow control, retransmission, credits, crc checking, message framing, and micropacket types all utilize the 32-bit control word associated with each micropacket. In addition to these mechanisms the data portion of the header micropacket contains information needed to steer a message through a switching fabric and to implement the semantics of message delivery between source and destination computer systems. The GSN header structure is depicted in Figure 5.

The first part of the header is called the MAC header since it resembles the headers found in other LAN architectures. The second part of the header micropacket is the IEEE 802.2 LLC/SNAP header and is used for packet demultiplexing. The remaining portion of the micropacket contains the payload. The MAC header and LLC/SNAP header are fixed in size and position as the first micropacket of any message. Message framing on GSN consists of locating a micropacket with TYPE=header and collecting the following TYPE=data micropackets until the Tail bit is observed.

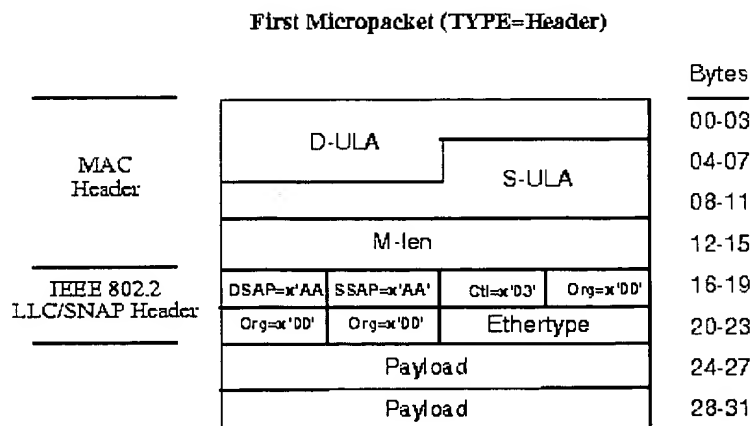


Figure 5

The MAC header contains a pair of 48-bit MAC addresses that indicate the destination and the source of the message. The address fields in the MAC header are intended for cut-through, or wormhole, packet switching. In this style of switch the initial micropacket establishes a connection between endpoints that exists only for the duration of the message transfer. The Tail bit of a message tears down the connection as it passes through the switching fabric.

M-len is a 32-bit field that indicates the message length: number of bytes in the message, not counting the header micropacket. The Protocol field (Proto) indicates the nature of the data payload in the message, i.e., a native GSN message, an Ethernet-style message, or a HIPPI-800 message. Other protocol types can be defined as needed. Bitflags are placed in the Flags field.

When the GSN Header micropacket is carrying an IP datagram (Ethertype=2048), the 8 bytes of payload in the Header micropacket are the first 8 bytes of the IP header. (Note that the 8 bytes immediately preceding the Payload are an 802.2 SNAP header.) When the GSN Header micropacket is carrying a scheduled (Ethertype=8181), the payload bytes in the Header micropacket are the initial 8 bytes of the ST Header.

Scheduled Transfers

Scheduled Transfer (ST) is an upper-layer protocol that can be implemented to operate over a number of physical layer subsystems, including GSN, HIPPI-800, ATM, and Ethernet. This section describes the main characteristics of the ST protocol. For the sake of introduction and ease of understanding, many of the less important functional details of ST are not covered in this description. Refer to the ANSI standards for complete details.

The most salient feature of ST is that it prepares both endpoints for the data movement before any data is transmitted. The first step in the preparation is to create a condition (state) called a virtual connection or VC. The second step is a handshake that allocates memory for the data movement and exposes this memory to the other endpoint. There are two flavors of the memory allocation handshake: one provides memory that is used once; the other provides memory that is used arbitrarily many times until released. The two endpoints exchange ST control operations to accomplish these pre-arrangements. Only after these prearrangements are complete can the first data movement begin; the data movement is performed with ST data operations.

GSN Resources

For general information and other resources: The High-Performance Networking Forum (HNF): www.hnf.org

For a Web Tour of GSN technology (courtesy of LANL): www.noc.lanl.gov/~jgd/hippi64/index.html

For related papers and links: CERN's GSN directory: www.cern.ch/HSI/hippi/gsn/gsnhome.htm#general

For GSN ANSI Standards Resources: www.cic-5.lanl.gov/lanp/ANSI/

For Scheduled Transfer Protocol ANSI Standard Resources: www.cic-5.lanl.gov/lanp/ANSI/cST.html

For European events and resources: The European High-performance-networking User Group (EHUG)

Footnotes

1. This technology is described in the Hot Interconnects 1996 Conference proceedings: "The Silicon Graphics SPIDER Chip" by Mike Galles. GSN is based on the LLP link technology used in SPIDER.
2. There are certain cases where the correct action is to forward the micropacket and set the ERROR bit.



Corporate Office
2011 N. Shoreline Boulevard
Mountain View, CA 94043
(650) 960-1980
URL: <http://www.sgi.com>

U.S. 1(800) 800-7441
Europe (44) 118-925.75.00
Asia Pacific (81) 3-54.88.18.11
Latin America 1(650) 933.46.37

Canada 1(905) 625-4747
Australia/New Zealand (61) 2.9879.95.00
SAARC/India (91) 11.621.13.55
Sub-Saharan Africa (27) 11.884.41.47